

1       **In the Claims**

2       Please cancel claims 25 and 32-36 without prejudice.

3       Claims 26, 28, 29, 37 and 38 are amended.

4       Claims 1-24, 26-31 and 37-39 remain in the application for consideration  
5 and are listed below:

6  
7       1.     (Original) An application program interface (API) comprising:  
8       a set of one or more residual difference data structures including residual  
9 difference information for encoded multimedia content; and  
10       a corresponding set of one or more control command data structures  
11 including control commands to control prediction and addition of residual coding  
12 information to decode multimedia content, wherein the API includes the control  
13 commands necessary to control multimedia decoding in accordance with any of a  
14 plurality of standard multimedia codecs.

15  
16       2.     (Original) An API according to claim 1, wherein the residual  
17 difference data structures and the control command data structures are two of a  
18 plurality of dynamically selected types of operational data structure(s) of the API.

19  
20       3.     (Original) An API according to claim 1, wherein the control  
21 command data structures include macroblock control commands, to control  
22 prediction and addition of residual coding information on a macroblock level.

1           4.     (Original) An API according to claim 1, wherein the API selectively  
2 invokes at least the subset of control commands upon identifying a processing  
3 capability of at least the decoder and the accelerator.

4  
5           5.     (Original) An API according to claim 4, wherein the API iteratively  
6 issues a set of control commands to negotiate an acceptable communication  
7 capability between one or more decoder applications and one or more hardware  
8 accelerators.

9  
10          6.     (Original) An API according to claim 1, wherein the API negotiates  
11 an acceptable communication capability between one or more decoders and one or  
12 more hardware accelerators by iteratively issuing configuration commands  
13 reflecting various alternative degrees and methods of decoding acceleration  
14 capability until choosing one that is acceptable to both the decoder(s) and the  
15 accelerator(s).

16  
17          7.     (Original) An API according to claim 1, further comprising:  
18 data structures, generated in response to command(s) received from a  
19 decoder application, consisting of deblocking filter control command(s) to control  
20 one or more deblocking filter attributes of a communicatively coupled hardware  
21 accelerator.

22  
23          8.     (Original) An API according to claim 7, wherein the deblocking  
24 filter control commands control a smoothing filter across block boundaries of a  
25 decoded picture.

1  
2 9. (Original) An API according to claim 7, wherein the deblocking  
3 filter control commands include a flag sent for each block edge denoting whether  
4 the deblocking filter is to be applied across the associated block edge.  
5

6 10. (Original) An API according to claim 1, wherein the control  
7 command data structure is a fixed-size data structure for each macroblock of a  
8 picture.  
9

10 11. (Original) An API according to claim 10, wherein the API utilizes an  
11 absolute macroblock address within each control command data structure to  
12 specify which macroblock to process, facilitating independent processing of each  
13 individual macroblock control command of a picture.  
14

15 12. (Original) An API according to claim 11, wherein the absolute  
16 macroblock address facilitates parallel processing of two or more macroblocks of  
17 a picture.  
18

19 13. (Original) An API according to claim 11, wherein the API utilizes a  
20 data location pointer within each control command data structure to specify the  
21 location within the corresponding residual difference data buffer for the data  
22 associated with the macroblock control command, facilitating independent  
23 processing of each individual macroblock control command of a picture.  
24  
25

1           14.    (Original) An API according to claim 13, wherein the data location  
2 pointer facilitates parallel processing of two or more macroblocks of a picture.

3  
4           15.    (Original) A storage medium comprising a plurality of executable  
5 instructions which, when executed, implement an application program interface  
6 (API) according to claim 1.

7  
8           16.    (Original) A computing system comprising:  
9 a storage medium including a plurality of executable instructions; and  
10 an execution unit, coupled to the storage medium, to execute at least a  
11 subset of the executable instructions to implement an application program  
12 interface (API) according to claim 1.

13  
14           17.    (Original) An application program interface (API) comprising:  
15 one or more auto-negotiation data structures, dynamically generated by the  
16 API to negotiate at least a set of processing standards among and between one or  
17 more elements of a media processing system; and  
18 one or more operational data structures, dynamically generated by the API  
19 to support processing of media content among and between the media processing  
20 system elements in accordance with the negotiated processing standard(s).

21  
22           18.    (Original) An API according to claim 17, the operational data  
23 structures comprising:  
24 a set of one or more residual difference data structures including residual  
25 difference information for encoded multimedia content; and

1 a set of one or more control command data structures including control  
2 commands to control prediction and addition of residual coding information to  
3 decode multimedia content, wherein the API includes the control commands  
4 necessary to control multimedia decoding in accordance with any of a plurality of  
5 multimedia codecs and invokes at least a subset of the control commands to  
6 interface the decoder with the multimedia accelerator.

7  
8 19. (Original) An API according to claim 17, the operational data  
9 structures comprising:

10 a raw bitstream data structure, dynamically generated to transfer raw media  
11 content bitstream(s) between media processing system elements.

12  
13 20. (Original) An API according to claim 17, wherein the auto-  
14 negotiation data structure(s) are dynamically generated to negotiate a split in  
15 media processing between identified media processing system elements.

16  
17 21. (Original) An API according to claim 20, the operational data  
18 structures comprising:

19 a set of one or more residual difference data structures including residual  
20 difference information for encoded multimedia content; and

21 a set of one or more control command data structures including control  
22 commands to control prediction and addition of residual coding information to  
23 decode multimedia content, wherein the API includes the control commands  
24 necessary to control multimedia decoding in accordance with any of a plurality of  
25

1 multimedia codecs and invokes at least a subset of the control commands to  
2 interface the decoder with the multimedia accelerator;

3 wherein the residual difference data structures and the control command  
4 data structures are dynamically generated to facilitate shared media processing  
5 between a decoder application executing on a host computer and a hardware  
6 accelerator, communicatively coupled to the host computer based, at least in part,  
7 on the auto-negotiation data structure.

8  
9 22. (Original) An API according to claim 20, the operational data  
10 structures comprising:

11 a raw bitstream data structure, dynamically generated to transfer raw media  
12 content bitstream(s) to facilitate media content decoding on a hardware accelerator  
13 communicatively coupled to a host computer implementing the API based, at least  
14 in part, on the auto-negotiation data structure.

15  
16 23. (Original) A storage medium comprising a plurality of executable  
17 instructions which, when executed, implement an API according to claim 17.

18  
19 24. (Original) A storage medium comprising a plurality of executable  
20 instructions which, when executed, implement an application program interface  
21 (API) to facilitate communication between elements of a media processing system,  
22 the API including one or more auto-negotiation data structures, dynamically  
23 generated by the API to negotiate at least a set of processing standards among and  
24 between one or more elements of a media processing system, and one or more  
25 operational data structures, dynamically generated by the API to support

1 processing of media content among and between the media processing system  
2 elements in accordance with the negotiated processing standard(s).

3  
4 25. (Canceled).

5  
6 26. (Currently Amended) ~~A method according to claim 25;~~ A method  
7 facilitating media processing between elements of a media processing system, the  
8 method comprising:

9 negotiating a media processing standard acceptable to each of the media  
10 processing system elements from a plurality of media processing standards; and  
11 dynamically generating operational data structures to support the negotiated  
12 media processing among and between the media processing system elements,

13 wherein negotiating a media processing standard comprises:

14 generating auto-negotiation data structure(s) configured in accordance with  
15 a proposed media processing standard;

16 issuing the auto-negotiation data structure(s) to each element of the media  
17 processing system; and

18 adopting the media processing system standard if each of the elements  
19 accept the proposed media processing standard.

20  
21 27. (Original) A method according to claim 26, further comprising:  
22 iteratively performing the generating and issuing steps utilizing a different  
23 proposed media processing standard on subsequent iterations until an acceptable  
24 media processing standard is adopted.  
25

1           28.    (Currently Amended) ~~A method according to claim 25,~~ A method  
2 facilitating media processing between elements of a media processing system, the  
3 method comprising:

4           negotiating a media processing standard acceptable to each of the media  
5 processing system elements from a plurality of media processing standards; and  
6           dynamically generating operational data structures to support the negotiated  
7 media processing among and between the media processing system elements,

8           wherein dynamically generating operational data structures comprises:  
9           generating residual difference data structure(s) to pass residual difference  
10 information between media processing system elements;

11           generating control command data structure(s) to pass control commands  
12 tailored in accordance with an adopted media processing standard based, at least in  
13 part, on the auto-negotiation.

14  
15           29.    (Currently Amended) ~~A method according to claim 25,~~ A method  
16 facilitating media processing between elements of a media processing system, the  
17 method comprising:

18           negotiating a media processing standard acceptable to each of the media  
19 processing system elements from a plurality of media processing standards; and  
20           dynamically generating operational data structures to support the negotiated  
21 media processing among and between the media processing system elements,

22           wherein the auto-negotiation data structure(s) also include a proposed split  
23 in media processing between the media processing system elements.



1           30.   (Original) A method according to claim 29, wherein dynamically  
2 generating operational data structures comprises:

3           generating raw bitstream data structure(s), to pass raw bitstream media  
4 content from a decoder application to a hardware accelerator to decode the media  
5 content based, at least in part, on the negotiated split in media processing between  
6 the decoder application and the hardware accelerator.

7  
8           31.   (Original) A method according to claim 29, wherein dynamically  
9 generating operational data structures comprises:

10          generating a residual difference data structure, to pass residual difference  
11 information between media processing system elements; and

12          generating a control command data structure, to pass control commands  
13 tailored in accordance with an adopted media processing standard;

14          wherein the residual difference data structure and the control command data  
15 structure are generated to facilitate shared decoding among two or more media  
16 processing system elements, as negotiated between the media processing system  
17 elements.

18  
19          32.   (Canceled).

20  
21          33.   (Canceled).

22  
23          34.   (Canceled).

24  
25          35.   (Canceled).

1  
2 36. (Canceled).

3  
4 37. (Currently Amended) ~~A computing system according to claim 36, A~~  
5 computing system comprising:

6 a plurality of non-integrated media processing system elements, to receive  
7 and process media content; and

8 an application program interface (API), communicatively coupling the non-  
9 integrated media processing system elements, to automatically negotiate a media  
10 processing standard acceptable to each of the plurality of non-integrated media  
11 processing system standards, and to dynamically generate data structures to  
12 facilitate media processing by the system elements, wherein the API dynamically  
13 generates a residual difference data structure to pass residual difference  
14 information, and a control command data structure to pass macroblock control  
15 commands tailored in accordance with an acceptable media processing system  
16 standard, when two or more media processing system elements decode the  
17 received media content,

18 wherein the API also negotiates a split in media processing among the  
19 media processing system elements based, at least in part, on an identified media  
20 processing capability of the media processing system elements.

21  
22 38. (Currently Amended) ~~A computing system according to claim 36, A~~  
23 computing system comprising:

24 a plurality of non-integrated media processing system elements, to receive  
25 and process media content; and

1        an application program interface (API), communicatively coupling the non-  
2 integrated media processing system elements, to automatically negotiate a media  
3 processing standard acceptable to each of the plurality of non-integrated media  
4 processing system standards, and to dynamically generate data structures to  
5 facilitate media processing by the system elements, wherein the API also  
6 negotiates a split in media processing among the media processing system  
7 elements based, at least in part, on an identified media processing capability of the  
8 media processing system elements, and wherein the API dynamically generates a  
9 raw bitstream data structure to pass raw bitstream media content from one media  
10 processing system element to another media processing system element to decode  
11 the received media content.

12  
13        39.    (Original) A computing system according to claim 38, wherein a  
14 decoder application passes received media content to one or more hardware  
15 accelerators to decode the media content via the raw bitstream data structure(s).